$2.50

April 1978 Volume 6 Number 4

Popular Computing

The only magazine in the world devoted to the art of computing

Annual Forecast For The Coming Decade In Computing

Texas Instruments enters the personal computer field (Sep 1978)

...r line of ...ing complex ...re exceeds Sep 1978)

A new largest prime number is found (Dec 1978)

Number of makers of large main frames drops to seven (Nov 1978)

PL/I is declared unofficially defunct (Nov 1978)

Number of home-built computers in the U.S. exceeds 35,000 (Aug 1978)

...ntries into ...l computing ...ut TI's 1979)

Computer kits disappear, replaced by prepackaged units (Feb 1979)

Users of microcomputers rediscover 3-address logic (Feb 1979)

Giant computer embezzlement scheme is revealed, involving government funds (Jan 1979)

Cost of executed instructions falls below 100 million per dollar (Jun 1979)

...es ...al ...uting ...d ...ops 8 (Dec 1979)

Pocket calculators with 10,000 program steps now cost $150 (Jun 1979)

Deliveries begin of cars having extensive computer controls on-board (Sep 1979)

The pocket calculator industry reaches 35 million units per year (Sep 1979)

Calendar

			1	2	3		4
	5	6	7		8	9	10
	11		12	13	14	15	
	16	17	18	19		20	21
	22	23		24	25	26	27
		28	29	30	31		32
	33	34	35		36	37	38
	39		40	41	42	43	
	44	45	46	47		48	49
	50	51		52	53	54	55
		56	57	58	59		60
	61	62	63		64	65	66
	67		68	69	70	71	
	72	73	74	75		76	77
	78	79		80	81	82	83
		84	85	86	87		88
	89	90	91		92	93	94
	95		96	97	98	99	
Jan	1	2	3	4	5	6	7
Feb	4	5	6	7	1	2	3
Mar	4	5	6	7	1	2	3
Apr	7	1	2	3	4	5	6
May	2	3	4	5	6	7	1
Jun	5	6	7	1	2	3	4
Jul	7	1	2	3	4	5	6
Aug	3	4	5	6	7	1	2
Sep	6	7	1	2	3	4	5
Oct	1	2	3	4	5	6	7
Nov	4	5	6	7	1	2	3
Dec	6	7	1	2	3	4	5
Jan, leap year	7	1	2	3	4	5	6
Feb, leap year	3	4	5	6	7	1	2

FOR THE 20th century: find the year at the top of this table. Follow down its column to the desired month, noting the exceptions for January and February of leap years; the number shown for that month indicates the corresponding calendar on the facing page. For the months of February, April, June, September, and November, delete the surplus days at the end of the month.

In this century, every year divisible by 4 is a leap year. Thus, the year 2000 will be a leap year, and it falls in the first column of the table.

[The table above, as well as the 7 month calendars, can be programmed as computer output, making a nice demonstration souvenir.]

Publisher: Audrey Gruenberger
Editor: Fred Gruenberger
Associate Editors: David Babcock
Irwin Greenwald
Contributing Editors: Richard Andree
William C. McGee
Thomas R. Parkin
Advertising Manager: Ken W. Sim
Art Director: John G. Scott
Business Manager: Ben Moore

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $20.50 per year, or $17.50 if remittance accompanies the order. For Canada and Mexico, add $1.50 per year. For all other countries, add $3.50 per year. Back issues $2.50 each.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

3

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

4

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

5

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

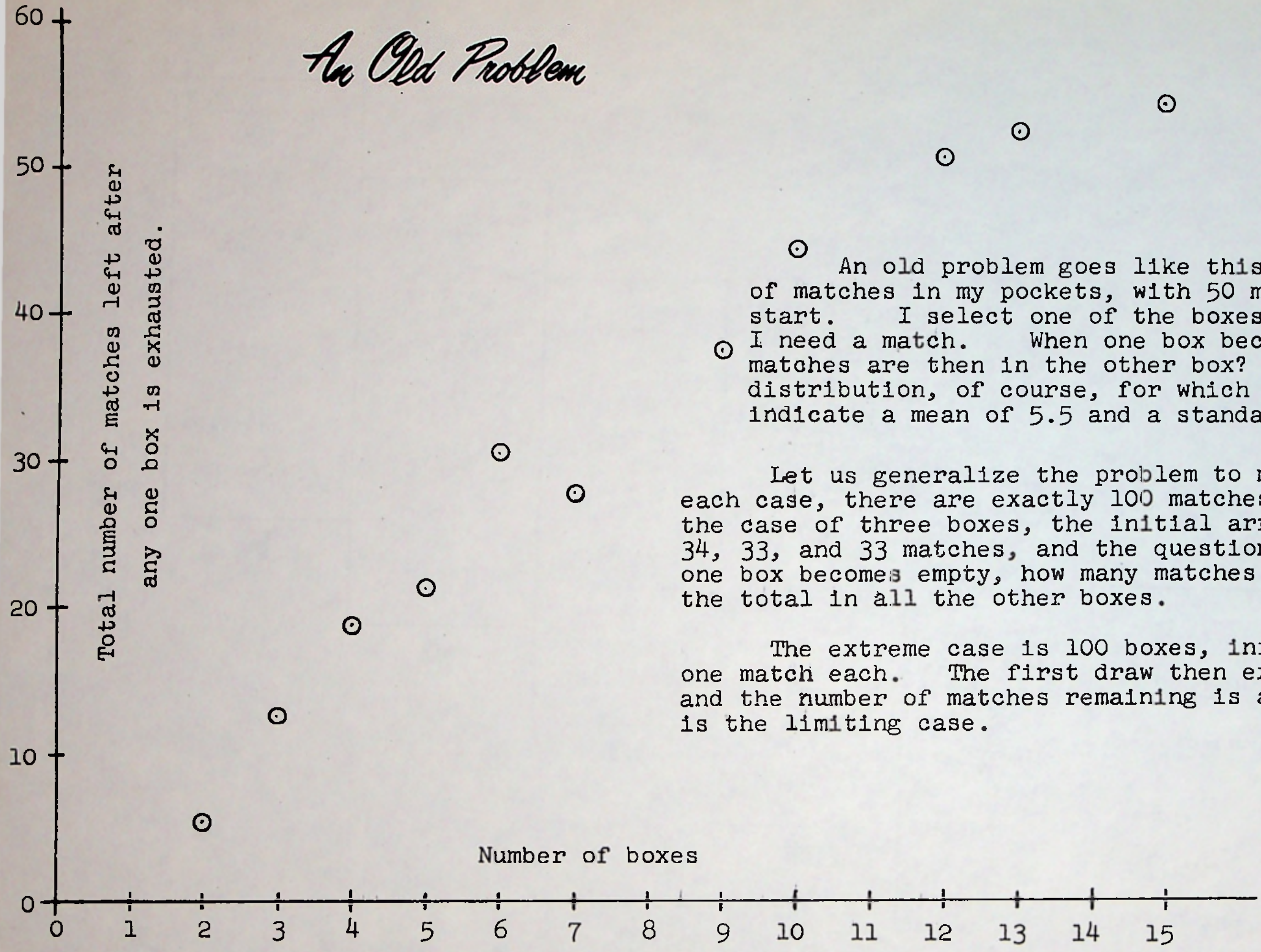
6

					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

7

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

An Old Problem



An old problem goes like this: I have two boxes of matches in my pockets, with 50 matches in each box to start. I select one of the boxes at random each time I need a match. When one box becomes empty, how many matches are then in the other box? The answer is a distribution, of course, for which some empirical runs indicate a mean of 5.5 and a standard deviation of 4.7.

Let us generalize the problem to more boxes. In each case, there are exactly 100 matches to start. For the case of three boxes, the initial arrangement is to have 34, 33, and 33 matches, and the question becomes: when one box becomes empty, how many matches are left; that is, the total in all the other boxes.

The extreme case is 100 boxes, initially containing one match each. The first draw then exhausts one box, and the number of matches remaining is always 99. This is the limiting case.

The interesting cases are those from 2 to 33 boxes. The accompanying graph shows some rough results. What is sought is more precise values and/or a formula for predicting the results for any case.

So you think you understand how random processes will behave?

PROBLEM **225**

BOOK REVIEW

THE STANDARD DATA ENCRYPTION ALGORITHM

by Harry Katzan, Jr.
Petrocelli Books, 1977, 134 pages, $12.

Security, privacy, and confidentiality are currently hot topics. One way to safeguard information is to encipher it--a task not well understood or performed by amateurs. Not only can the task be bungled (leading to a false sense of security) but if done badly can consume inordinate amounts of CPU time. For many such reasons, plus the advantages accruing from standardization, the National Bureau of Standards has sought, certified, and promulgated a standard encryption algorithm (intended to be implemented in special purpose hardware). The details of the logic of the NBS algorithm constitute the subject of this book.

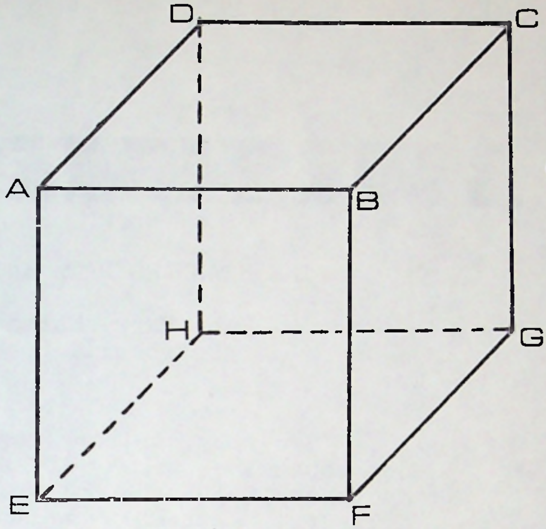
Inasmuch as the NBS standard was published January 15, 1977, a book appearing in 1977 to explain it ought to find a lively market.

But not a book as hastily rushed into production as this one. While it might be possible to unravel the details of the algorithm from this book, the task would be far easier if one consulted the NBS document itself (Data Encryption Standard, FIPS Publication 46, U.S. Department of Commerce) or one of Herb Bright's papers ("Cryptography Using Modular Software Elements," in the Proceedings of the 1976 National Computer Conference, AFIPS Volume 45, pp. 113-123) or IBM documents on the subject (IBM is the primary vendor of hardware for the NBS algorithm).

This book begins with a discussion of the problem of safeguarding information (if you weren't aware of the problem, you wouldn't be reading this book). Then follows a chapter on elementary cryptography, about at the level of the Sunday papers, or General Mills' magic decoder ring (except for two paragraphs on Lester Hill's landmark work on the application of simultaneous linear equations to cryptographic work).

It is a poor book, rushed into print to beat out potential competitors (the spellings "permuted" and "permutted" are used with equal frequency, dozens of times). It may indicate, however, a potential market for a more careful treatment of the subject. It works out here to over 20¢ per useful page; at that price, no published book can compete with copying machines.

VERTEX NUMBERING



The 8 vertices of a cube are numbered from 1 to 8. Each of the 12 edges is then assigned a value which is the absolute value of the difference between the vertex numbers at its ends. The sum of the edge numbers varies with the ordering of the vertices.

With the scheme shown here for ordering the vertices of the cube, we want the sum of the absolute values of:

A - B	B - C	C - D	D - A
A - E	B - F	C - G	D - H
E - F	F - G	G - H	H - E.

This sum could be 28 (numbering the vertices 12345678) or it could be 44 (with the numbering 16378542). Larger or smaller sums may be possible. It should not be difficult to find the largest and smallest possible sums, inasmuch as there are only 2520 ways in which the vertices of the cube can be numbered from 1 to 8.

When the problem is extended to a dodecahedron, however, it becomes more interesting. A regular dodecahedron has 12 faces, each a regular pentagon; it has 20 vertices and 30 edges.

The following is suggested as a scheme for ordering the vertices. Select one face and letter the five vertices A, B, C, D, and E. Position the dodecahedron so that that face is uppermost. Then letter the vertex directly below A as F, and letter the vertices around the equator from F through Ø (thus there will be edges AF, BH, CJ, DL, and EN). The vertex just below Ø becomes P, and the bottom pentagon is lettered from P through T, so that there are edges ØP, GQ, IR, KS, and MT.

What ordering of the numbers from 1 to 20 on the vertices will produce

(a) maximum
(b) minimum

sums of the edge values, where each edge's value is the absolute difference of the vertex numbers at its ends?

NUMBER TRIVIA

		0		1		2		3		4		5		6		7		8		9
10	2	101	2	103	3	1065	3	1095	3	1125	3	1160	3	1195	3	1230	3	1260	3	1300
11	2	122	2	124	2	126	2	128	5	2	2	133	2	135	2	137	4	2	2	142
12	2	146	2	147	2	152	2	153	5	3	3	2	2	159	2	162	2	164	2	167
13	2	170	4	3	2	175	2	178	2	180	2	183	2	185	5	5	2	191	2	194
14	2	197	2	2	2	202	5	6	3	3	2	211	2	214	5	7	2	220	4	5
15	2	226	5	8	2	232	2	235	2	238	5	9	4	6	2	247	3	4	2	253
16	2	257	5	11	4	7	2	266	5	12	2	273	2	276	5	13	4	8	5	14
17	3	5	5	15	6	26	2	3	5	16	6	29	5	17	4	10	5	18	6	33
18	5	19	3	6	4	11	5	21	6	39	5	22	4	12	5	23	5	24	4	13
19	5	25	3	7	6	51	4	14	5	28	6	55	4	15	5	30	5	31	6	63
20	6	65	5	33	5	34	4	17	5	36	5	37	5	38	6	79	3	9	5	40
21	5	41	4	20	5	43	5	44	4	21	3	10	4	22	5	49	4	23	5	51
22	5	52	4	24	3	11	2	5	5	57	5	58	5	59	4	27	3	12	5	63
23	4	28	5	66	5	68	5	69	5	71	3	13	5	74	5	75	5	77	4	33
24	5	80	3	14	5	83	5	85	2	6	5	89	3	15	5	92	4	38	5	96
25	5	98	3	16	5	102	4	41	5	106	5	108	5	110	3	17	5	115	4	45
26	4	46	5	122	3	18	5	126	2	7	5	131	5	134	4	51	5	139	5	141
27	5	144	4	54	5	149	5	152	5	155	3	21	5	161	4	59	5	167	5	170
28	4	62	5	176	2	8	5	182	5	185	4	66	5	192	5	195	3	24	4	70
29	5	206	4	72	5	213	5	216	5	220	5	224	4	77	5	232	5	236	5	239
30	4	82	5	250	3	28	5	259	5	262	5	265	5	271	4	89	5	280	4	92
31	5	289	6	905	5	298	6	941	3	31	5	313	2	10	5	322	4	103	5	331
32	4	105	5	343	5	349	5	352	5	358	5	364	4	114	3	35	5	382	4	118
33	4	120	2	11	5	406	3	37	4	125	5	424	5	430	5	436	5	442	3	39
34	5	457	5	463	4	137	5	481	4	141	5	499	2	12	5	508	5	517	4	149
35	3	43	5	535	4	154	4	156	5	562	5	571	4	162	5	580	3	46	5	598
36	4	168	5	616	5	625	4	174	5	643	3	49	5	661	4	182	5	679	4	186
37	4	188	4	190	4	192	4	194	4	196	4	198	4	200	4	203	4	205	4	207
38	3	55	4	212	4	213	4	216	4	218	4	220	4	222	4	226	4	228	4	230
39	4	232	4	234	4	238	3	61	4	241	4	245	4	247	4	250	4	252	4	255

The table on these three pages shows a list of irrational numbers (2nd, 3rd, 4th, 5th, and 6th roots) whose leading digits are those indexed. For example, the first entry indicates that the square root of 101 has the leading digits 101; the tenth entry indicates that the cube root of 1300 has the leading digits 109, and so on.

		0		1		2		3		4		5		6		7		8		9
40	4	257	5	1041	3	65	5	1068	6	4352	5	1095	6	4502	5	1122	3	68	5	1149
41	6	4772	5	1176	6	4892	5	1203	3	71	4	297	5	1257	3	73	5	1284	4	309
42	5	1311	5	1338	4	318	3	76	4	324	3	77	5	1419	4	333	5	1446	3	79
43	5	1473	5	1500	4	349	5	1527	5	1554	4	359	5	1581	5	1608	4	369	5	1635
44	5	1662	3	86	5	1689	5	1716	5	1743	4	393	5	1770	5	1797	3	90	4	407
45	5	1851	4	414	4	418	4	424	5	1932	4	430	4	433	5	2013	4	442	5	2040
46	5	2067	3	98	5	2121	5	2148	3	100	5	2175	4	472	4	478	5	2256	5	2283
47	5	2310	5	2337	5	2364	3	106	5	2418	4	511	5	2445	3	109	5	2499	5	2526
48	5	2553	4	538	4	541	4	547	4	550	4	556	4	559	4	565	4	568	4	574
49	3	118	4	583	4	586	4	592	4	598	4	601	4	607	4	611	4	619	4	621
50	4	628	3	126	3	127	3	128	4	646	3	129	3	130	3	131	4	666	3	132
51	4	677	4	687	3	135	5	3575	5	3600	4	707	5	3675	5	3700	5	3750	2	27
52	5	3825	4	937	5	3900	5	3950	4	757	3	145	5	4050	5	4075	5	4125	2	28
53	5	4200	4	797	5	4275	5	4325	4	817	5	4400	5	4425	3	155	2	29	5	4550
54	5	4600	4	857	3	160	4	870	3	161	4	887	3	163	2	30	3	165	3	166
55	4	917	3	168	3	169	4	937	4	947	3	171	2	31	4	967	3	174	4	977
56	4	987	3	177	3	178	4	1007	3	180	2	32	4	1027	3	183	4	1047	3	185
57	4	1057	3	187	3	188	3	189	2	33	4	1097	3	192	3	193	4	1117	4	1127
58	4	1137	4	1147	3	198	2	34	4	1167	3	201	4	1187	3	203	3	204	4	1207
59	3	206	2	35	3	208	3	209	4	1247	3	211	4	1267	3	213	3	214	3	215
60	4	1297	3	218	4	1317	3	220	4	1337	3	222	4	1356	3	224	2	37	3	226
61	4	1387	3	229	4	1407	3	231	4	1427	3	233	2	38	4	1457	3	237	4	1477
62	3	239	3	240	3	241	3	242	2	39	4	1527	3	246	3	247	3	248	3	250
63	4	1577	3	252	2	40	3	254	4	1617	3	257	3	258	4	1647	3	260	4	1677
64	2	41	3	264	3	265	4	1717	3	268	3	269	3	270	4	1757	3	273	3	274
65	4	1787	4	1797	4	1817	3	279	4	1837	2	43	3	283	4	1867	3	285	4	1887
66	4	1907	3	290	3	291	3	292	3	293	4	1967	3	296	3	297	3	299	4	2007
67	2	45	3	303	4	2047	3	305	3	307	3	308	3	310	3	311	3	312	3	314
68	3	315	3	316	3	318	3	320	3	321	2	47	3	323	3	325	4	2247	3	328
69	4	2267	3	330	2	48	3	333	3	335	3	336	3	338	3	340	3	341	3	342

		0		1		2		3		4		5		6		7		8		9
70	4	2407	3	345	3	346	4	2447	3	350	3	351	3	352	2	50	3	355	3	357
71	3	358	3	360	3	361	3	363	2	51	3	366	3	368	3	370	3	371	3	372
72	3	374	2	52	3	377	4	2747	3	380	3	382	4	2787	3	385	2	53	4	2827
73	3	390	3	391	3	393	3	395	2	54	3	398	3	400	3	401	3	402	4	2987
74	3	406	2	55	3	410	3	411	4	3067	3	415	3	416	3	417	2	56	3	421
75	3	422	4	3187	3	426	3	427	2	57	3	431	4	3267	3	435	4	3307	3	438
76	4	3347	2	58	4	3387	3	445	3	446	4	3427	3	450	3	452	3	453	4	3507
77	3	457	3	460	3	461	3	462	2	60	3	466	4	3627	3	470	3	472	3	473
78	4	3707	2	61	3	480	3	481	3	482	3	485	3	486	2	62	3	490	3	492
79	4	3907	3	495	3	497	2	63	3	501	3	503	4	4027	3	507	3	510	3	511
80	4	4107	3	515	3	516	3	518	3	520	3	522	2	65	3	527	3	528	3	530
81	4	4307	3	534	2	66	4	4387	3	540	4	4427	3	545	3	546	3	548	4	4507
82	3	552	3	555	3	556	3	558	2	68	3	562	4	4667	3	566	4	4707	3	570
83	2	69	4	4787	3	576	4	4827	3	581	3	583	2	70	3	587	4	4947	3	591
84	4	4987	3	595	2	71	4	5067	3	602	3	605	3	606	4	5147	2	72	3	612
85	4	5227	3	617	3	620	3	621	2	73	3	626	3	628	3	630	3	633	3	635
86	2	74	4	5507	3	642	3	643	3	645	3	648	2	75	3	652	3	655	4	5707
87	3	660	2	76	4	5787	3	666	3	668	4	5867	3	673	2	77	3	677	3	680
88	3	682	4	6027	3	687	2	78	3	691	3	695	4	6187	3	700	2	79	3	703
89	3	705	4	6307	3	710	4	6387	2	80	3	717	4	6467	3	723	3	725	3	727
90	4	6587	3	732	3	735	3	737	4	6707	2	82	3	745	3	747	4	6827	3	753
91	4	6867	2	83	3	760	3	763	4	6987	3	768	2	84	3	772	4	7107	4	7147
92	3	780	2	85	3	785	3	788	4	7307	3	793	3	795	2	86	4	7427	3	804
93	4	7507	3	807	2	87	3	813	3	815	3	819	4	7707	3	825	2	88	4	7787
94	3	832	3	835	3	838	2	89	3	842	3	845	3	847	3	850	2	90	3	855
95	3	860	3	861	3	865	2	91	3	870	3	873	3	875	3	879	3	880	2	92
96	3	885	3	890	3	891	3	895	2	93	3	900	3	903	3	905	3	909	2	94
97	3	915	3	918	3	920	3	924	2	95	3	927	3	930	3	935	3	936	2	96
98	3	942	3	945	3	948	3	950	2	97	3	957	3	960	3	963	3	965	2	98
99	3	972	3	975	3	978	3	980	2	99	3	987	3	990	3	993	3	995	3	999

Binary Curriculum

by John Motil

There is a time for recursion and a time for iteration
a time for top-down and a time for bottom-up
a time for software and a time for hardware
a time for batch processing and for time sharing
a time for the theoretical and also the practical
a time for efficiency and a time for beauty
a time for micro-machines and also monster-machines
a time for data structures and also program structures
a time for trees and a time for bits
a time for analysis and a time for synthesis
a time for privacy and a time for sharing
a time for speed and a time for space
a time for finite-state automata and also Turing Machines
a time for proof and a time for intuition
a time for the past and a time for the future
a time for determinism and a time for probability
a time for the instantaneous and for the sequential
a time for broad surveys and a time for deep research
a time for compilers and a time for interpretors
a place for early binding and a place for late binding
a time for fortran and a time for lisp
a time for flowcharts and a time for pseudo-code
a time for people and a time for machines
a time for logic and a time for engineering
a time for economics and a time for physics
but, there is no time for the GOTO.

... and there's a time for :
pushes and a time for pops
a time for compressing and a time for expanding
a time to sort and a time to randomize
a time to search and a time to hide ?
a time to enter and a time to exit
a time for series and a time for parallel
a time for special purpose and a time for general purpose
a time for fixed word lengths and a time for variable lengths
a time for prefix and a time for postfix
a time for the "while" and a time for the "until"
a time for induction and a time for deduction
a time to merge and a time to partition
a time for voltage and a time for current
a time to call by name and a time to call by value
a time to parse and a time to generate
a time for analog and a time for digital
a time for static (binding or allocation) and also dynamic
a time for ROMs and a time for RAMs.
a time to peek and a time to poke
a time to think and a time to run
a time for IBM and a time for everyone else
a time to buy and a time to sell
a time for micros and a time for macros ?
.... and there's a time
for all good things
to come to an end.

Congestion

Nine markers start on cell 00 in the accompanying diagram. At the time of a move, each marker advances a specific number of cells:

Marker	A	advances	19 cells
	B		23
	C		29
	D		31
	E		37
	F		41
	G		43
	H		47
	J		53

For example, if marker B is on cell 17, it will move to cell 40 at the time of a move. The diagram shows where the markers will be after ten moves. After

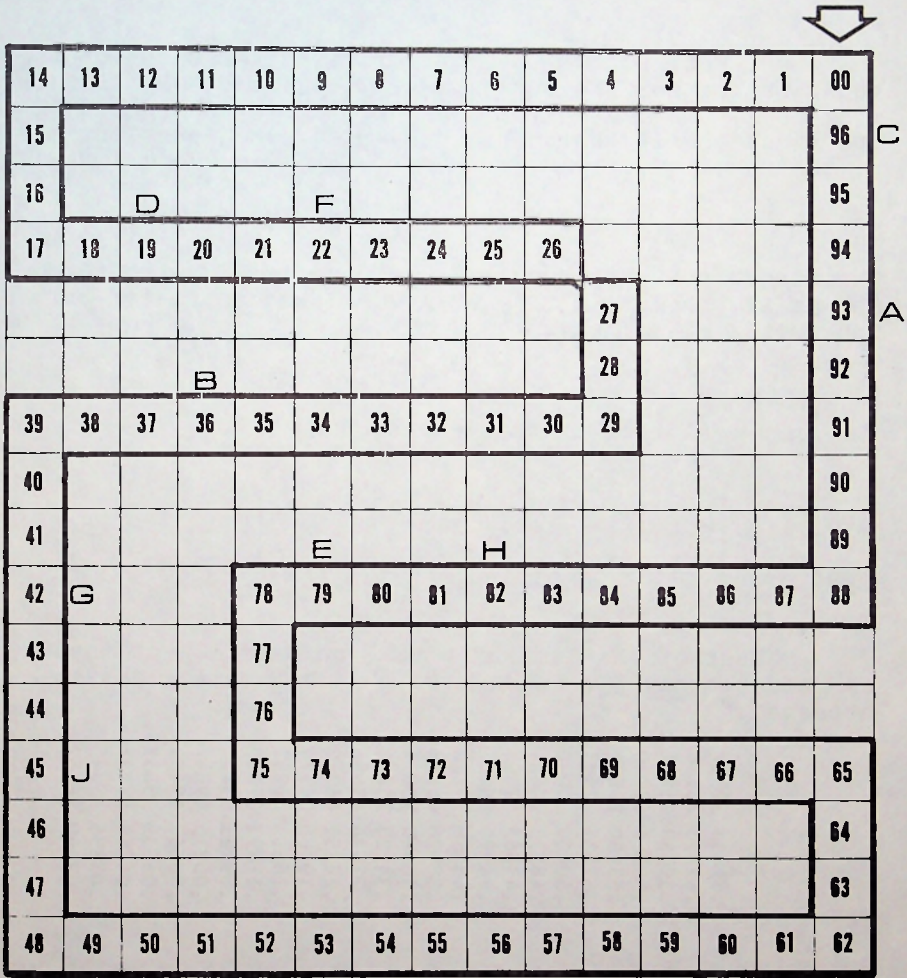
$$19 \times 23 \times 29 \times 31 \times 37 \times 41 \times 43 \times 47 \times 53$$

moves, all the markers would again fall on cell 00. Actually, all the markers will be on cell 00 after 97 moves, and any multiple of 97 moves thereafter.

It is reasonable to suppose that some congestion (say, 5 or more markers) will occur on some cell at other times, and it should be possible to calculate when and where this might occur. We might make this into a real computing problem with the following perturbation: when any marker falls on a cell numbered 80 or higher, it is to move one more cell higher. Thus, if marker J moves from cell 38, it would normally go to cell 91, but the extra rule moves it to cell 92.

With this new rule, find the move number and cell number whenever any move puts 5 or more markers on one cell.

CONGESTION



14 13 12 11 10 9 8 7 6 5 4 3 2 1 00
15 96 C
16 D F 95
17 18 19 20 21 22 23 24 25 26 94
27 93 A
B 28 92
39 38 37 36 35 34 33 32 31 30 29 91
40 90
41 E H 89
42 G 78 79 80 81 82 83 84 85 86 87 88
43 77
44 76
45 J 75 74 73 72 71 70 69 68 67 66 65
46 64
47 63
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62

Problem Solution

More information on problems 218 and 223 (Fun With Equations) is furnished by Herman P. Robinson, Lafayette, California.

For the equation

$$Ax^3 + x^2 + x - 1 = 0$$

we start with A = 1, find the positive real root, multiply that root by some factor F, and replace A by the product of F and the root. The process now repeats, and will converge. Problem 223 was to find the relation between the F values and the convergent roots.

Mr. Robinson writes: "After many iterations of the equation the coefficient of x^3 will be, say, C. Let the root be x, and multiply it by F giving xF, the new C. But since the iteration has converged, the equations must be the same, or C = xF. Then

$$Fx^4 + x^2 + x - 1 = 0.$$

The single positive root of this equation furnishes the answer to problem 223. When F = 1 we can factor out the root x = -1, leaving

$$x^3 - x^2 + 2x - 1 = 0.$$

The only real root of this equation is:

$$1/3 + (1/3)\left[(11/2) + (3/2)\sqrt{69}\right]^{1/3} + (1/3)\left[(11/2) - (3/2)\sqrt{69}\right]^{1/3}$$

$$= .56984\ 02909\ 98053\ 26591\ 13999\ 58119\ 56864\ 88397\ 97439\ 12894$$

"Most of the list of x vs F on page 10 (of issue 58) is good to only 5 or 6 digits. A corrected list is attached."

F	x
0	.61803 39887
0.1	.61175 28184
0.5	.59050 66741
1	.56984 02910
2	.53950 22822
3	.51737 37437
4	.50000 00000
5	.48573 30424
6	.47365 52088
9	.44579 85267
10	.43840 34211
20	.38926 46413
30	.36088 75874
40	.34123 25604
50	.32634 85056
100	.28257 46212
500	.19766 34774
1000	.16835 26952
5000	.11491 37030
10000	.09721 93827
20000	.08215 49047
50000	.06567 19654
100000	.05539 35833
200000	.04669 83392
1000000	.03136 38846
2000000	.02640 94152

Problem Solution

Problem number 25 (The Four Subroutines) appeared in issue No. 9. There are four subroutines given, each of which outputs a stream of non-decreasing integers, one integer per call of the subroutine. The four subroutines are independent of each other. The output of all four of the subroutines is to be written out to tape in ascending order with no duplicates.

Gary Laughton, of the University of California at San Diego, furnishes the flowchart given here. The four subroutines are each called once, furnishing numbers called A, B, C, and D. The word called OUT is set to zero. The logic of the merge then begins at Reference 1 of Mr. Laughton's flowchart.

The problem also called for a test procedure. Mr. Laughton suggests the following:

> Arrange to have subroutine A generate odd numbers, starting with 1.
> Have subroutine B furnish powers of 2, starting with 2.
> Let subroutine C generate the Fibonacci sequence, starting with 1, 1, 2, 3,...
> Let subroutine D generate squares, starting with 1.

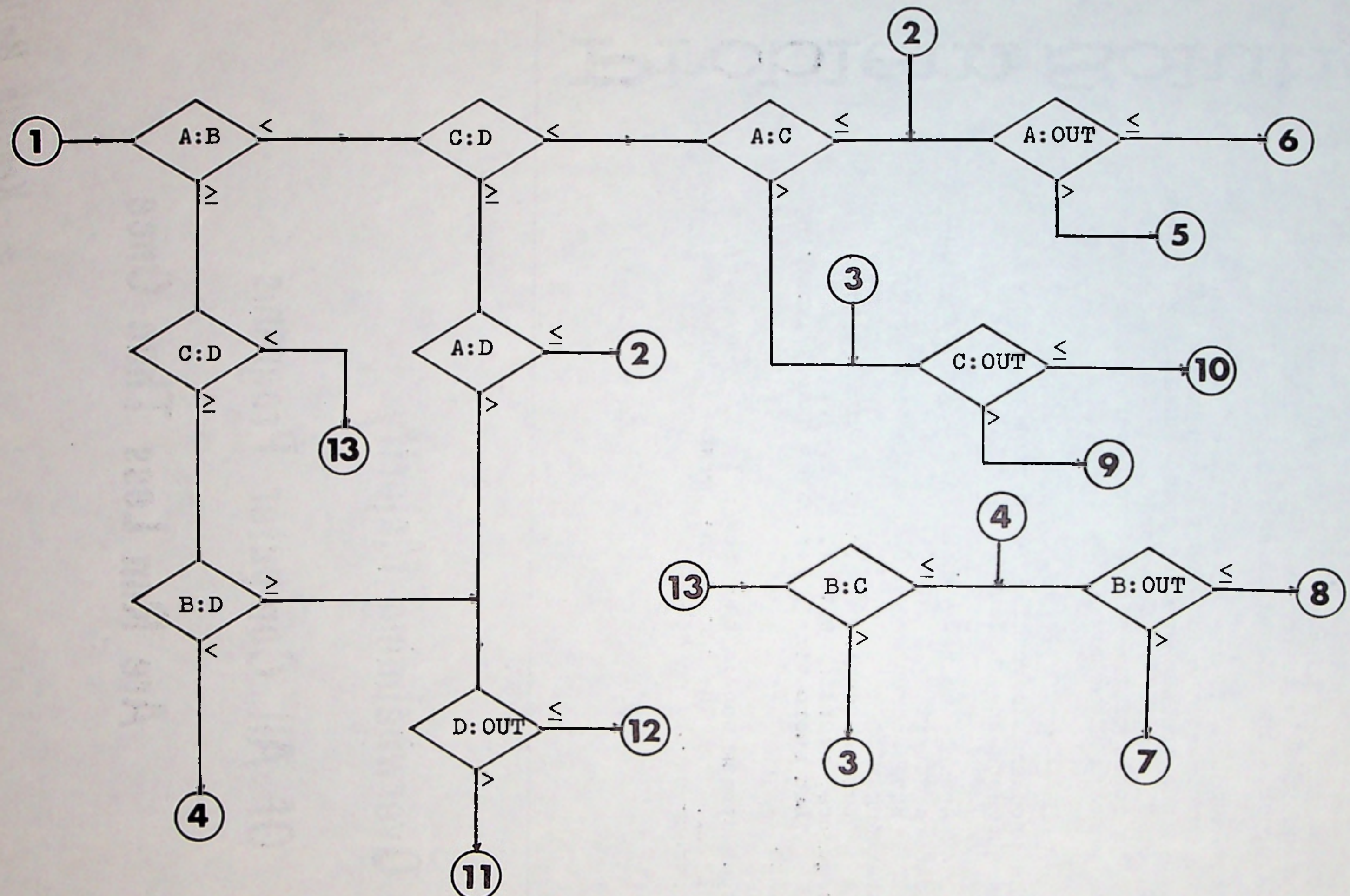
The output stream should then begin 1, 2, 3, 4, 5, 7, 8, 9, 11, 13,... The sum of the first 25 numbers in the output stream should be 456.

The Overwhelming Majority

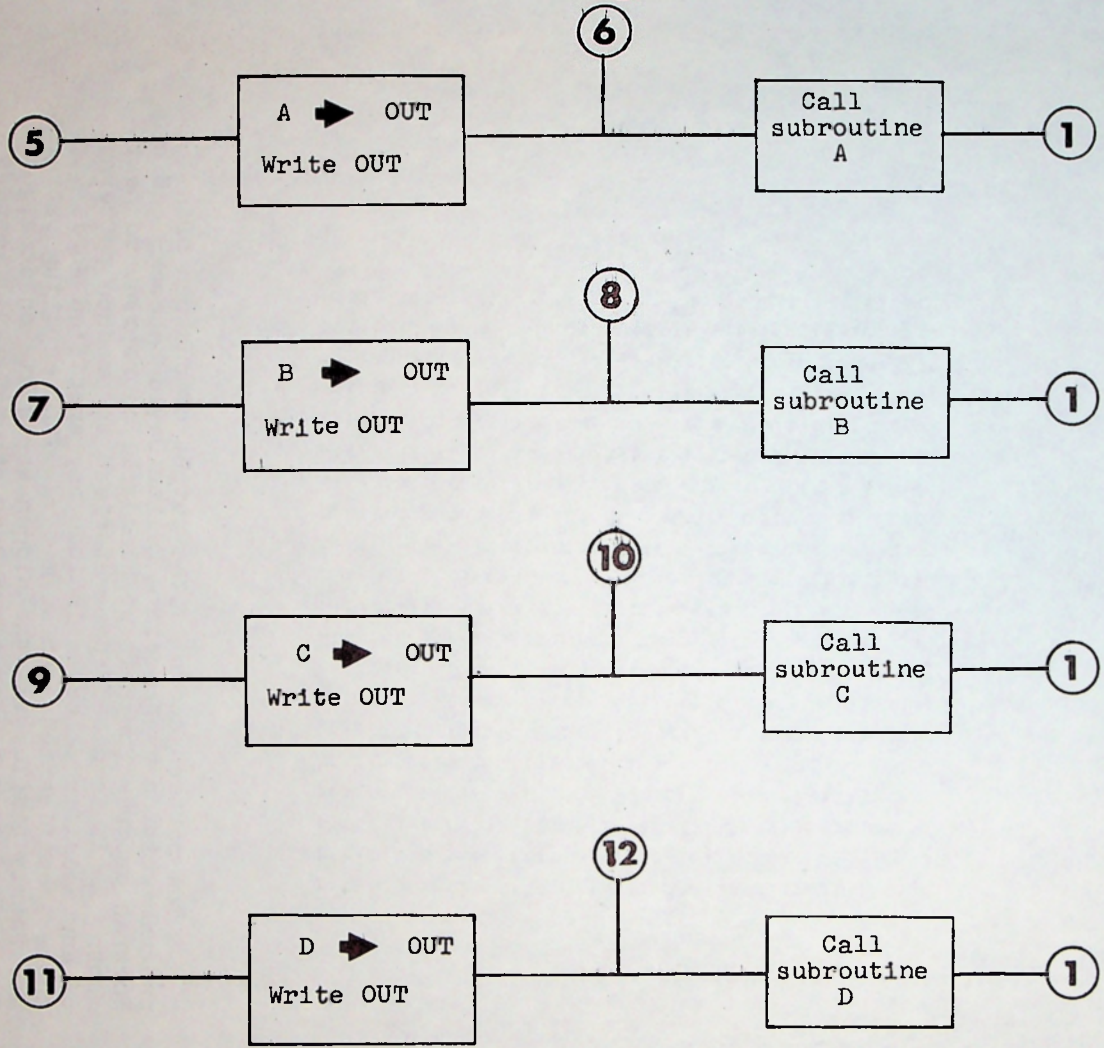
Of All Computer Programs

Are Run Less Than Once

--Kenneth Boilen



A possible solution to the Four Subroutines Problem of issue No. 9. The suggested test procedure will produce the number 103 as the 61st number of the output stream, and the sum up to that point is 3000.


6
5
A → OUT
Write OUT
Call subroutine A
1
8
7
B → OUT
Write OUT
Call subroutine B
1
10
9
C → OUT
Write OUT
Call subroutine C
1
12
11
D → OUT
Write OUT
Call subroutine D
1

For The Kiddies

The pattern shown below is a familiar puzzle; namely, to find the buried names by tracing patterns horizontally or vertically or diagonally through the seemingly scrambled letters. There are 35 names of 33 notable people in computing in this pattern. It is mildly amusing to trace out the 34 names in addition to the one that has been highlighted.

```
W U D O J X I S K Z J M B K B A C K U S
F P O D P N P Y F Y V H J A O Z B J U O
G I R N N A M U E N N O V I B U T J P G
B I N D H O P P E R D A D A R C X H V N
R S G C E K V D P F L H A D M A O F Y I
O G T D Y E K S U H Z N L B G S O C W R
O N S H A W A K Z L S K X M E R Z E K U
K I R Z T L O H H C U B C K R F C S I T
S T C O N G R O S C H C L E C A Y T T A
F S C T S Q E L A I R I S Q L Z B I T Y
P A A U P E S U Z A W T H E S L T B R L
E H G N E K I A C H E T V K L R N I E H
W U N B S I N K F R I O R E A B D T K G
J X I A Z X E O Z R L D H U P O S Z C U
M H M B L N C S E N N S W A T S O N E A
K Z M B E G S L R E G R E B N E U R G M
G Y A A K Y L D W G I U P R P G I L L J
B Z H G F O S G E A Y K X S B Z M Y H K
S N P E H E S R I O D V A R T S K J I D
X E W I Z A J Y N G E T A M R N W C V A
R W V U M O G I B T K Y R A C O U G D R
Y E O R D R W Z E E E I L N J M K E L A
B L K N U T H C R Z A K I J D I O I L R
Z L T X N X V Z G I A F H S T S F K K Y
```

The pattern shown is the output of one run of a program written by Associate Editor David Babcock. His Fortran program for this problem will be reproduced next month. It is nicely parameterized to produce patterns of any specified size to include any specified words (if those specifications can be met). The resulting patterns are fairly simple (all the hidden words lie on straight lines, and they seldom cross each other). All letters not used by the hidden words are selected at random.

With the same pattern size and the same hidden words, any number of different patterns can be created. Thus, for example, at Christmas time a unique pattern can be made for each child in a class, with all the Christmas words (SNOW, SANTA, SLEIGH, etc.) hidden.

Even with the restriction that the hidden words lie along straight lines, the patterns could become much more complex by arranging to have the words intersect, as the nine words do in this pattern:

```
R P D C H R I S T M A S
D E L V E S A U F R Q M
O F E A R B K A S W T V
E Q M D I B T L A W E F
S S T Y N G E C N R O S
H L M A G I L H T A M B
T P C N G F E R A D F H
I F W H E T H R W R L Q
G N W O N S S M B K R H
```

And finally, if the hidden words were allowed to go around corners, the resulting patterns could become quite challenging. We would welcome other programs that produce such interesting patterns.

ADA	BUCHHOLTZ	GRUENBERGER	LOVELACE	STIBITZ
AIKEN	CARY	HAMMING	MAUCHLY	TURING
AMDAHL	DIJKSTRA	HASTINGS	MCCRACKEN	VONNEUMANN
BABBAGE	ECKERT	HOLLERITH	NEWELL	WATSON
BABCOCK	FORRESTER	HOPPER	SHAW	WEINBERG
BACKUS	GILL	HUSKEY	SHELL	WILKES
BROOKS	GROSCH	KNUTH	SIMON	ZUSE

Computing courses are required in more than half of all U.S. high schools (Apr 1982)

More than half of all U.S. computing power is now residing in small dedicated machines (Mar 1981)

Number of persons employed full time as "programmers" starts to drop (Jun 1981)

A new "new largest prime number" (this one of 9000 digits) is found (Mar 1980)

Cost of executed instructions falls below one billion per dollar (Dec 1981)

Personal computers with power exceeding that of a 7094 now are available for under $2000, including peripherals (Jun 1982)

Restrictive legislation on the us data banks is enacted (Jan 1983)

Superprogrammers now rate $50K salaries (Dec 1980)

The job category "programmer" has disappeared (Nov 1983)

The sixth prime repunit number is discovered by an undergraduate student (Aug 1978)

Cost p produc softw 0

The last traditional textbook publisher gives up (Jun 1985)

Cost per line of producing complex software exceeds $100 (Mar 1984)

Standard word length on personal computers is 32 bits (Jun 1983)

Japanese the perso field unde prices (Apr

Top speeds of supercomputers reach one nanosecond complete floating add time (Jul 1985)

Hard copy mail falls below 50% of 1977 level (Feb 1986)

Number of makers of large main frames drops to six (Sep 1987)

Number of centralized large computers in the U.S. falls below 1000 (May 1987)

Number magazi in th rs com fi t

All distinctions between micro and mini computers have disappeared (Jun 1987)